

LAPORAN
PRAKTIKUM PEMROGRAMAN WEB

Jobsheet-06: PHP02

Dosen Pengampu:
Dimas Wahyu Wibowo, ST., MT.



Disusun oleh:
Siti Fa'iqoh 2241760026

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2024



Topik

1. konsep *function* pada pemrograman PHP
2. konsep penggabungan pada pemrograman PHP dan HTML

Tujuan

Mahasiswa diharapkan mampu:

1. Mahasiswa paham dengan konsep *function* pada pemrograman PHP
2. Mahasiswa paham dengan konsep penggabungan pada pemrograman PHP dan HTML

Pendahuluan

Pengenalan Array dan Fungsi

Array atau larik adalah salah satu tipe data. Array bukan merupakan tipe data dasar seperti integer atau boolean, tetapi array adalah sebuah tipe data yang terdiri dari kumpulan tipe data lainnya. Array memudahkan dalam membuat kelompok data, menghemat penulisan, dan penggunaan variabel. Di dalam PHP terdapat 3 jenis array, yaitu *indexed array*, *associative array*, dan *multidimensional array*.

Selain tipe data, pemahaman tentang fungsi adalah salah satu yang dibutuhkan ketika membuat program. Dalam bahasa pemrograman, fungsi didefinisikan sebagai kode program yang dirancang untuk menyelesaikan sebuah tugas tertentu, dan merupakan bagian dari program utama. Ketika merancang kode program kadang kita sering membuat kode yang melakukan tugas yang sama secara berulang-ulang, seperti membaca tabel dari database, menampilkan penjumlahan, dan lain-lain. Tugas yang sama ini akan lebih efektif jika dipisahkan dari program utama, dan dirancang menjadi sebuah fungsi.

Praktikum Bagian 1. *Indexed Array*

Indexed array adalah array dengan indeks numerik. Penulisan *indexed array* bisa dilakukan secara otomatis atau diberikan indeks secara manual. Indeks array selalu dimulai dari 0. Berikut adalah cara penulisan *indexed array*:

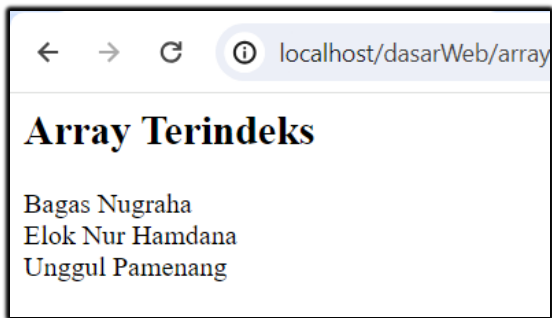
```
<?php
    $variable = array(
        "value0",
        "value1",
        "value2",
        :
        "value-n"
    );
?>
```

atau bisa juga dituliskan seperti berikut:

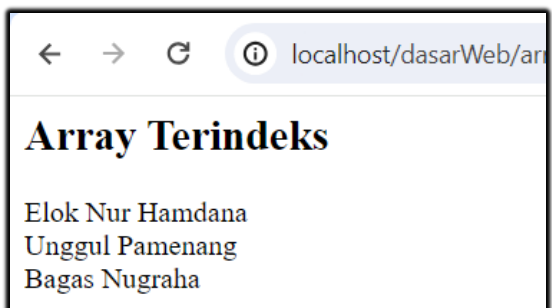
```
<?php
    $variable[0] = "value0";
    $variable[1] = "value1";
    $variable[2] = "value2";
    :
    $variable[n] = "value-n";
?>
```

Ikuti langkah-langkah berikut untuk memahami *indexed* array di dalam PHP:

Langkah	Keterangan
1	<p>Buat file baru dengan nama <code>array_1.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre><!DOCTYPE html> <html> <head> </head> <body> <h2>Array Terindeks</h2> <?php \$Listdosen=["Elok Nur Hamdana","Unggul Pamenang", "Bagas Nugraha"]; echo \$Listdosen[2] . "
"; echo \$Listdosen[0] . "
"; echo \$Listdosen[1] . "
"; ?> </body> </html></pre>
2	Simpan file dan jalankan kode program dengan mengetikkan <code>localhost/dasarWeb/praktik_php/array_1.php</code>
3	Amati hasil yang ditampilkan
4	<p>Untuk menampilkan array, selain menggunakan indeks kita juga bisa menggunakan perulangan. Coba tampilan hasil dari kode program diatas dengan menggunakan perulangan. Tampilkan kode program dan hasilnya. (soal no.1)</p> <pre>C: > xampp > htdocs > dasarWeb > array_1.php > html 1 <html> 2 <head></head> 3 <body> 4 <h2>Array Terindeks</h2> 5 <?php 6 \$Listdosen=["Elok Nur Hamdana","Unggul Pamenang","Bagas Nugraha"]; 7 8 echo \$Listdosen[2]."
"; 9 echo \$Listdosen[0]."
"; 10 echo \$Listdosen[1]."
"; 11 12 // foreach (\$Listdosen as \$dosen) { 13 // echo \$dosen . "
"; 14 // } 15 16 ?> 17 </body> 18 </html></pre>



```
C: > xampp > htdocs > dasarWeb > array_1.php > html
1  <html>
2  <head></head>
3  <body>
4  <h2>Array Terindeks</h2>
5  <?php
6  $Listdosen=["Elok Nur Hamdana","Unggul Pamenang","Bagas Nugraha"];
7
8  // echo $Listdosen[2]."<br>";
9  // echo $Listdosen[0]."<br>";
10 // echo $Listdosen[1]."<br>";
11
12 foreach ($Listdosen as $dosen) {
13     echo $dosen . "<br>";
14 }
15
16 ?>
17 </body>
18 </html>
```



array akan menyimpan nama-nama dosen tersebut, dan pemanggilan elemen array yaitu dimulai dari indeks ke-0 untuk elemen pertama.

Praktikum Bagian 2. Associative Array

Komponen *associative* array terdiri dari pasangan kunci (*key*) dan nilai (*value*). Kunci menunjukkan posisi dimana nilai disimpan. PHP menggunakan tanda panah (*=>*) untuk mendefinisikan nilai kepada kunci. Berikut adalah cara penulisan *associative* array:

```
<?php
$variable = array(
    "Key0" => "value0",
    "key1" => "value1",
    "key2" => "value2",
    :
    "key-n" => "value-n"
);

?>
```

atau bisa juga ditulis seperti berikut:

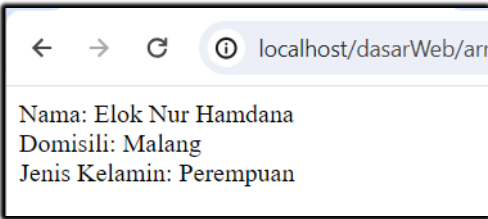
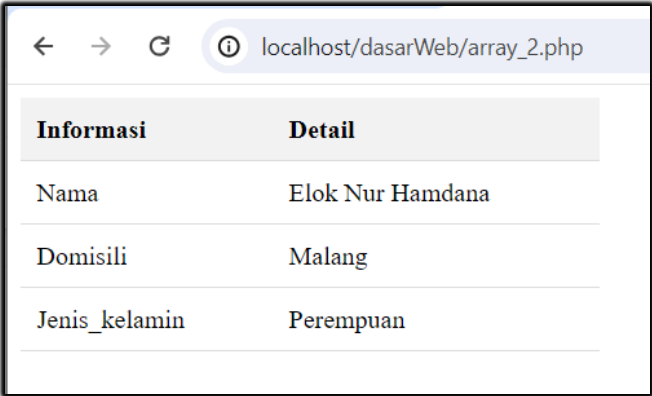
```
<?php
$variable['key0'] = "value0";
```

```

$variable['key1'] = "value1";
$variable['key2'] = "value2";
:
$variable['key-n'] = "value-n";
?>

```

Ikuti langkah-langkah berikut untuk memahami *associative* array di dalam PHP:

Langkah	Keterangan
1	<p>Buat file baru dengan nama <code>array_2.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre> <!DOCTYPE html> <html> <head> <meta charset="utf-8"> <meta name="viewport" content="width=device-width, initial-scale=1"> <title></title> </head> <body> <?php \$Dosen = ['nama' => 'Elok Nur Hamdana', 'domisili' => 'Malang', 'jenis_kelamin' => 'Perempuan']; echo "Nama : {\$Dosen ['nama']}
"; echo "Domisili : {\$Dosen ['domisili']}
"; echo "Jenis Kelamin : {\$Dosen ['jenis_kelamin']}
"; ?> </body> </html> </pre>
2	<p>Simpan file dan jalankan kode program dengan mengetikkan <code>localhost/dasarWeb/praktik_php/array_2.php</code></p>
3	<p>Amati hasil yang ditampilkan. Kemudian tambahkan style tabel pada output tampilan tersebut supaya lebih menarik. (soal no.2)</p> <p>*Untuk penggunaan style bebas boleh internal atau eksternal file.</p> <div>   </div> <p>Berdasarkan hasil dan kode diatas kita bisa menggunakan array asosiatif, sehingga kita bisa memanggil array dengan nama variabelnya</p>

Praktikum Bagian 3. *Multidimensional Array*

Multidimensional array adalah array yang terdiri dari satu atau lebih array. Berikut adalah contoh penulisan array dengan dimensi 2:

```
<?php
    $variable = array(
        array ("value00", "value01", .., "value-nm"),
        array ("value10", "value11", .., "value-nm"),
        array ("value20", "value21", .., "value-nm"),
    );
?>
```

Ikuti langkah-langkah berikut untuk memahami multidimensional array di dalam PHP:

Langkah	Keterangan
1	<p>Buat file baru dengan nama <code>style.css</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre>1 table { 2 border-collapse: collapse; 3 border-spacing: 0; 4 width: 100%; 5 border: 1px solid #ddd; 6 } 7 8 th, td { 9 text-align: left; 10 padding: 16px; 11 } 12 13 tr:nth-child(even) { 14 background-color: #f2f2f2; 15 }</pre>
2	<p>Buat file baru dengan nama <code>array_3.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre>1 <!DOCTYPE HTML> 2 <html> 3 <head> 4 <link rel="stylesheet" type="text/css" href="style.css"/> 5 </head> 6 <body> 7 <h2> Multidimensional Array </h2> 8 <table> 9 <tr> 10 <th>Judul Film</th> 11 <th>Tahun</th> 12 <th>Rating</th> 13 </tr> 14 <?php 15 \$movie = array(16 array("Avengers: Invinity War", 2018, 8.7), 17 array("The Avengers", 2012, 8.1), 18 array("Guardians of the Galaxy", 2014, 8.1), 19 array("Iron Man", 2008, 7.9) 20); 21 echo "<tr>"; 22 echo "<td>". \$movie[0][0] . "</td>"; 23 echo "<td>". \$movie[0][1] . "</td>"; 24 echo "<td>". \$movie[0][2] . "</td>"; 25 echo "</tr>"; 26 echo "<tr>"; 27 echo "<td>". \$movie[1][0] . "</td>"; 28 echo "<td>". \$movie[1][1] . "</td>"; 29 echo "<td>". \$movie[1][2] . "</td>"; 30 echo "</tr>";</pre>

	<pre> 31 echo "<tr>"; 32 echo "<td>". \$movie[2][0] . "</td>"; 33 echo "<td>". \$movie[2][1] . "</td>"; 34 echo "<td>". \$movie[2][2] . "</td>"; 35 echo "</tr>"; 36 echo "<tr>"; 37 echo "<td>". \$movie[3][0] . "</td>"; 38 echo "<td>". \$movie[3][1] . "</td>"; 39 echo "<td>". \$movie[3][2] . "</td>"; 40 echo "</tr>"; 41 ?> 42 </table> 43 </body> 44 </html> </pre>
3	Simpan file dan jalankan kode program dengan mengetikkan localhost/dasarWeb/praktik_php/array_3.php
4	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 3)</p> <p>Dengan menggunakan array multidimensi, seperti dalam contoh movie[0][0] untuk judul film, movie[0][1] untuk tahun, dan movie[0][2] untuk rating, kita dapat menyimpan banyak elemen dalam satu struktur data yang terorganisir. Ini memungkinkan penyimpanan dan pengelolaan data yang lebih efisien dan terstruktur.</p>

Fungsi

Banyak fungsi *build-in* dari php yang sering kita gunakan, seperti `print()`, `print_r()`, `unset()`, dll. Selain fungsi-fungsi tersebut, kita juga dapat membuat fungsi sendiri sesuai kebutuhan.

Fungsi adalah sekumpulan intruksi yang dibungkus dalam sebuah blok. Fungsi dapat digunakan ulang tanpa harus menulis ulang instruksi di dalamnya.

Fungsi pada PHP dapat dibuat dngan kata kunci *function*, lalu diikuti dengan nama fungsinya.

Contoh:

```

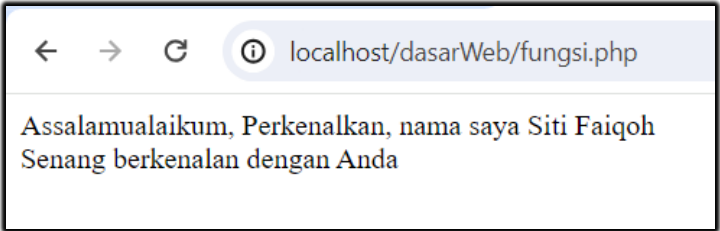
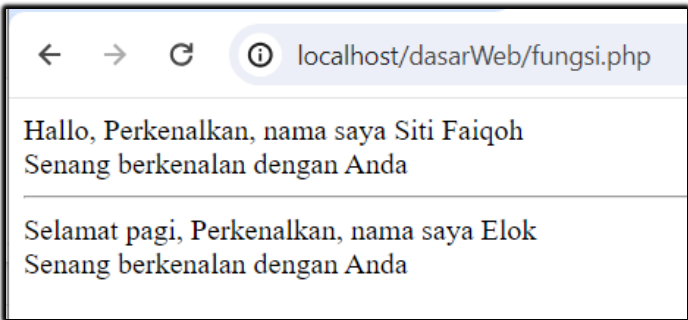
function namaFungsi(){
    //...
}

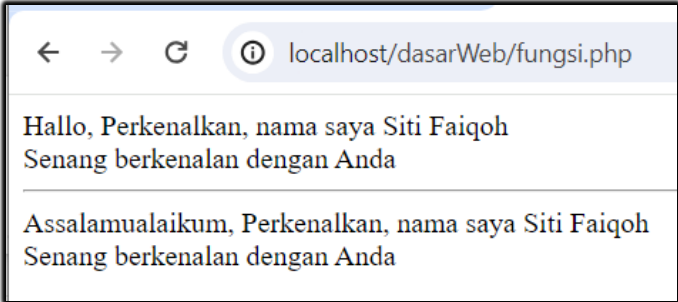
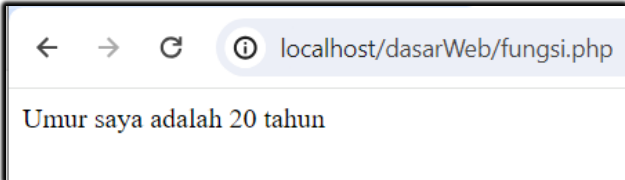
```

Kode instruksi dapat di tulis di dalam kurung kurawal ({...})

Penamaan fungsi di dalam PHP diawali dengan huruf atau tanda garis bawah dan tidak boleh diawali dengan angka. Penulisan nama fungsi ini tidak **case-sensitive**. Ikuti langkah-langkah berikut untuk memahami penggunaan fungsi di dalam PHP:

Langkah	Keterangan
1	<p>Buatlah satu file baru di dalam direktori praktik_php , beri nama fungsi.php</p> <pre> <?php function perkenalan(){ echo "Assalamualaikum, "; echo "Perkenalkan, nama saya Elok
"; //Tulis sesuai nama kalian echo "Senang berkenalan dengan Anda
"; } //memanggil fungsi yang sudah dibuat perkenalan(); ?> </pre>
2	<p>Simpan file dan jalankan kode program dengan cetak sebanyak 2 kali, Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 4)</p>

	 <p>Fungsi pengenalan berguna untuk mempersingkat penulisan script. Sehingga ketika kita menuliskan pengenalan(), semua yang ada dalam fungsi pengenalan() akan ditampilkan.</p>
Fungsi dengan Parameter	
3	<p>Supaya instruksi yang di dalam fungsi lebih dinamis, kita dapat menggunakan parameter untuk memasukkan sebuah nilai ke dalam fungsi. Nilai tersebut akan diolah di dalam fungsi. Misalkan, pada contoh fungsi yang tadi, tidak mungkin nama yang dicetak adalah <i>elok</i> saja dan salam yang dipakai tidak selalu <i>assalamualaikum</i>.</p>
4	<p>Tambahkan parameter seperti pada kode program berikut ini:</p> <pre data-bbox="284 748 858 1218"><?php //membuat fungsi function perkenalan(\$nama, \$salam){ echo \$salam.", "; echo "Perkenalkan, nama saya ".\$nama."
"; echo "Senang berkenalan dengan Anda
"; } //memanggil fungsi yang sudah dibuat perkenalan("Hamdana","Hallo"); echo "<hr>"; \$saya = "Elok"; \$ucapanSalam = "Selamat pagi"; //memanggil lagi perkenalan(\$saya,\$ucapanSalam); ?></pre>
5	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 5)</p>  <p>Ketika kita memberikan parameter kedalam fungsi perkenalan(nama,salam), maka input tersebutlah yang akan dikeluarkan pada parameter yang berada di statement echo;</p>
Parameter dengan Nilai Default	
6	<p>Nilai <i>default</i> dapat kita berikan di parameter. Nilai <i>default</i> berfungsi untuk mengisi nilai sebuah parameter, kalau parameter tersebut tidak diisi nilainya. Misalnya: lupa mengisi parameter salam, maka program akan <i>error</i>. Oleh karena itu, kita perlu memberikan nilai <i>default</i> supaya tidak error.</p>
7	<p>Ketikkan kode program berikut</p>

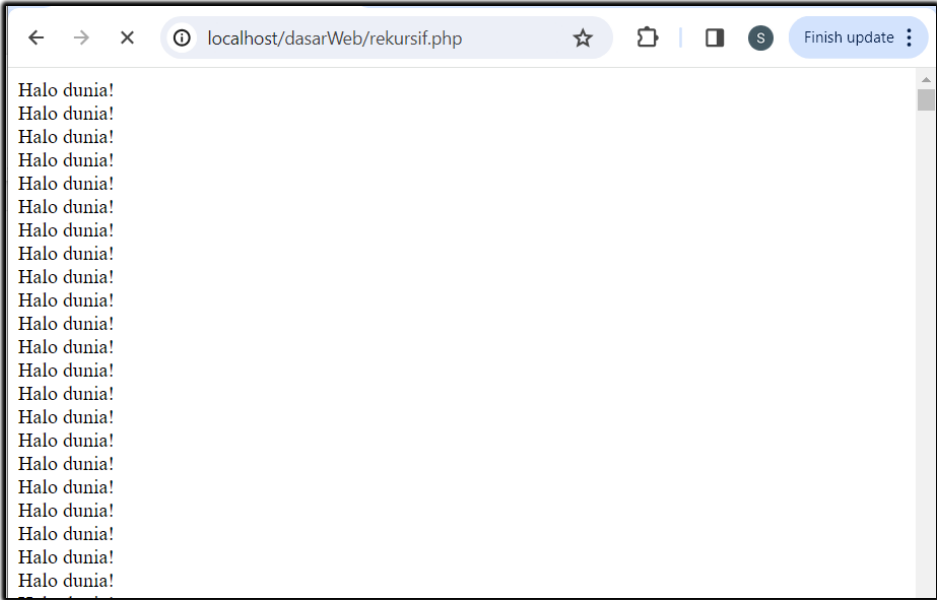
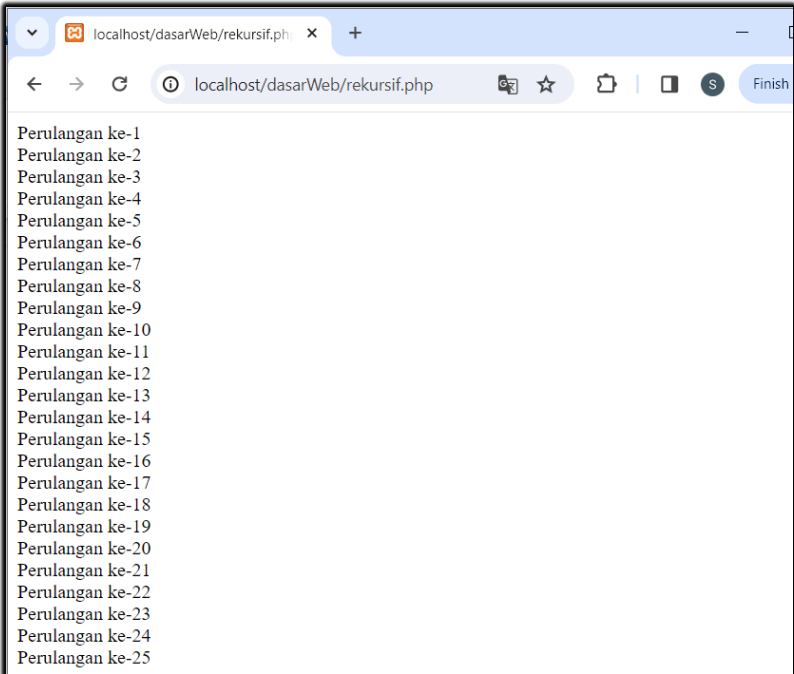
	<pre> <?php //membuat fungsi function perkenalan(\$nama, \$salam="Assalamualaikum"){ echo \$salam.", "; echo "Perkenalkan, nama saya ".\$nama."
"; echo "Senang berkenalan dengan Anda
"; } //memanggil fungsi yang sudah dibuat perkenalan("Hamdana","Hallo"); echo "<hr>"; \$saya = "Elok"; \$ucapanSalam = "Selamat pagi"; //memanggil lagi tanpa mengisi parameter salam perkenalan(\$saya); ?> </pre>
8	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 6)</p>  <p>pada pemanggilan fungsi dengan parameter default, ketika kita tidak menuliskan param ke dua, maka akan digantikan dengan parameter default yang sudah dituliskan di awal.</p>
Fungsi yang Mengembalikan Nilai	
9	<p>Hasil pengolahan nilai dari fungsi mungkin saja kita butuhkan untuk pemrosesan berikutnya. Oleh karena itu, kita harus membuat fungsi yang dapat mengembalikan nilai.</p> <p>Pengembalian nilai dalam fungsi dapat menggunakan kata kunci return.</p>
10	<p>Ketikkan kode program berikut</p> <pre> <?php //membuat fungsi function hitungUmur(\$thn_lahir, \$thn_sekarang){ \$umur = \$thn_sekarang - \$thn_lahir; return \$umur; } echo "Umur saya adalah ". hitungUmur(1988, 2023) ."tahun" // isi sesuai dengan tahun lahir kalian ?> </pre>
11	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 7)</p>  <p>Fungsi hitungUmur(2004, 2024) akan menghitung umur berdasarkan tahun lahir 2004 dan tahun sekarang 2024, dan kemudian mencetak hasilnya dengan kalimat "Umur saya adalah ... tahun"</p>
Memanggil Fungsi di dalam fungsi	
12	<p>Fungsi yang sudah kita buat, dapat juga dipanggil di dalam fungsi lain.</p>

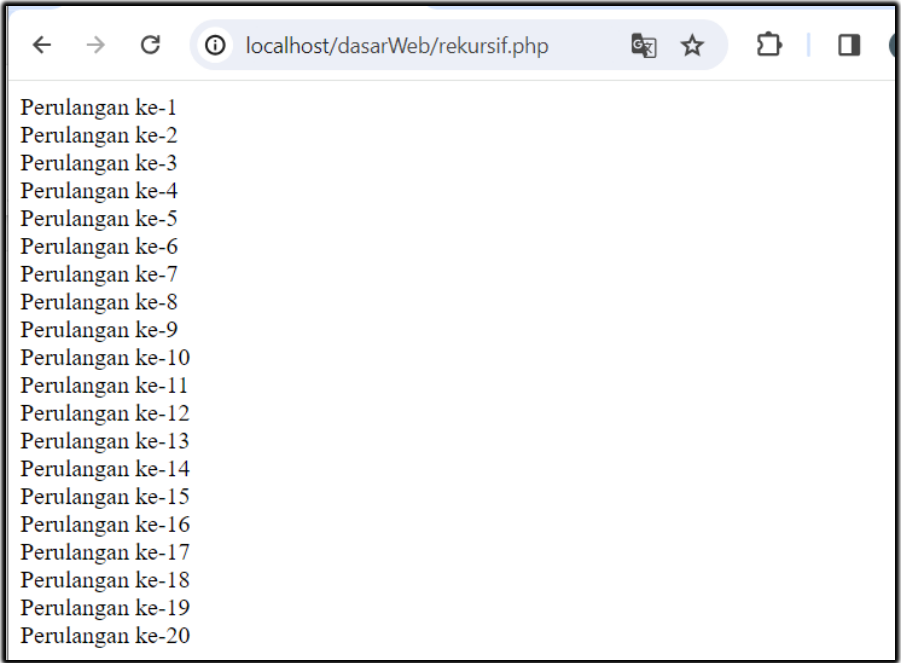
13	<p>Ketikkan kode program berikut</p> <pre><?php //membuat fungsi function hitungUmur(\$thn_lahir, \$thn_sekarang){ \$umur = \$thn_sekarang - \$thn_lahir; return \$umur; } function perkenalan (\$nama, \$salam="Assalamualaikum") { echo \$salam.", "; echo "Perkenalkan, nama saya ".\$nama."
"; //memanggil fungsi lain echo "Saya berusia ". hitungUmur(1988, 2023) ." tahun
"; echo "Senang berkenalan dengan anda
"; } //memanggil fungsi perkenalan perkenalan ("Elok"); ?></pre>
14	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 8)</p>  <p>Fungsi hitungUmur(\$thn_lahir, \$thn_sekarang) digunakan untuk menghitung umur berdasarkan tahun kelahiran dan tahun sekarang. Sedangkan fungsi perkenalan(\$nama, \$salam = "Assalamualaikum") mencetak pesan perkenalan dengan salam, nama, dan umur yang dihitung menggunakan fungsi hitungUmur().</p>

Fungsi Rekursif

Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri. Fungsi ini biasanya digunakan untuk menyelesaikan masalah seperti faktorial, bilangan fibbonaci, pemrograman dinamis

Langkah	Keterangan
1	<p>Buat file baru dengan nama rekursif.php di dalam direktori praktik_php, kemudian ketikkan kode berikut:</p> <pre><?php function tampilkanHaloDunia(){ echo "Halo dunia!
"; tampilkanHaloDunia(); } tampilkanHaloDunia(); ?></pre>
2	<p>Jika kode program di atas dijalankan, apa yang akan terjadi dan bagaimana dampaknya jika itu di lakukan kemukakan pendapat kalian! (soal no 9)</p>

	 <p>Tanpa penanganan untuk rekursi yang tidak berakhir, kode tersebut dapat menyebabkan terjadinya stack overflow. Hal ini disebabkan karena fungsi tampilkanHaloDunia() memanggil dirinya sendiri tanpa kondisi terminasi yang jelas. Akibatnya, pesan "Halo dunia!" akan terus dicetak tanpa henti sampai terjadi kesalahan stack overflow.</p>
3	<p>Untuk menampilkan angka 1 sampai 25, kita bisa dengan mudah menggunakan perulangan for seperti berikut:</p> <pre data-bbox="276 1003 823 1193"><?php for (\$i=1; \$i <=25; \$i++){ echo "Perulangan ke-{\$i}
"; } ?></pre> 
4	<p>Akan tetapi jika kita ingin menggunakan konsep fungsi rekursif untuk menjalankan tugas yang sama, ketikkan kode program berikut.</p>

	<pre> <?php function tampilkanAngka (int \$jumlah, int \$indeks = 1) { echo "Perulangan ke-{\$indeks}
"; //panggil diri sendiri selama \$indeks <= \$jumlah if (\$indeks < \$jumlah) { tampilkanAngka(\$jumlah, \$indeks + 1); } } tampilkanAngka(20); ?> </pre>
5	<p>Jalankan kode program di atas dan bagaimana outputnya kemudian jelaskan kenapa bisa seperti itu. (soal no 10)</p>  <p>Fungsi tampilkanAngka() akan menampilkan pesan "Perulangan ke-X" sebanyak 20 kali, di mana X adalah nilai iterasi dari variabel \$indeks. Fungsi ini menerima dua parameter, yaitu \$jumlah yang menentukan jumlah perulangan dan \$indeks yang merupakan nilai iterasi saat ini.</p>

Contoh Kasus Menu Bertingkat

Langkah	Keterangan
1	<p>Buat variabel \$menu. Variable ini adalah gabungan antara <i>array</i> terindeks dan <i>array</i> asosiatif multidimensi. Dikatakan multidimensi karena ia adalah suatu <i>array</i> yang memiliki array lain di dalamnya.</p> <p>Selanjutnya kita akan coba menampilkan semua item dari <i>array</i> \$menu menggunakan fungsi rekursif.</p>
2	Buatlah kode program untuk variabel \$menu berikut

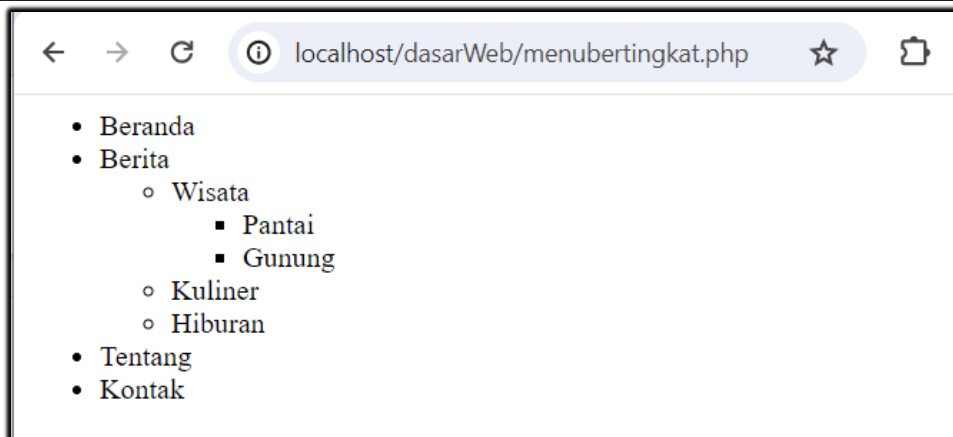
	<pre> <?php \$menu = [["nama" => "Beranda"], ["nama" => "Berita", "subMenu" => [["nama" => "Wisata", "subMenu" => [["nama" => "Pantai"], ["nama" => "Gunung"]]], ["nama" => "Kuliner"], ["nama" => "Hiburan"]]], ["nama" => "Tentang"], ["nama" => "Kontak"],]; </pre>
3	<p>Kemudian buatlah fungsi untuk menampilkan <i>array</i> utama</p> <pre> function tampilkanMenuBertingkat (array \$menu) { echo ""; foreach (\$menu as \$key => \$item) { echo "{\$item['nama']}"; } echo ""; } tampilkanMenuBertingkat(\$menu); ?> </pre>
4	<p>Jalankan program diatas dan bagaimana hasil outputnya (soal no 11)</p>  <p>Saat program dijalankan akan menampilkan menu bertingkat (hierarchical menu) dalam bentuk unordered list (HTML dan)</p>
5	<p>Selanjutnya buatlah fungsi di atas menjadi rekursif dengan memanggil dirinya sendiri ketika suatu item dari menu memiliki attribut subMenu. Sehingga tampilanya menjadi seperti berikut. (soal no 12)</p>

- Beranda
- Berita
 - Wisata
 - Pantai
 - Gunung
 - Kuliner
 - Hiburan
- Tentang
- Kontak

```

46 // soal no 12
47 function tampilkanMenuBertingkat(array $menu) {
48     echo "<ul>";
49     foreach ($menu as $item) {
50         echo "<li>{$item['nama']}</li>";
51         if (isset($item['subMenu']) && is_array($item['subMenu'])) {
52             tampilkanMenuBertingkat($item['subMenu']);
53         }
54     }
55     echo "</ul>";
56 }
57 tampilkanMenuBertingkat($menu);
58
59 ?>

```



Program menampilkan menu bertingkat dengan tambahan kode `isset($item['subMenu']) && is_array($item['subMenu'])` dan pemanggilan rekursif `tampilkanMenuBertingkat($item['subMenu'])`. Ini memeriksa apakah setiap item dalam menu memiliki submenu (subMenu) dan apakah submenu tersebut merupakan array. Jika ya, submenu tersebut akan ditampilkan secara rekursif, membangun menu bertingkat dengan semua submenu yang ada.

String

String tipe data karakter yang biasanya diekspresikan dengan diapit oleh tanda petik ganda (“ ”) atau petik tunggal (‘ ’). Keduanya hampir sama akan tetapi cara kerjanya sedikit berbeda. Tanda petik dua akan mengisi variabel dengan nilai aslinya, ada pun dengan tanda petik satu, kita tidak bisa melakukan hal tersebut.

Contoh strings adalah “Hello world!”. Beberapa operasi dapat dilakukan pada data bertipe string. PHP menyediakan fungsi-fungsi *built-in* yang siap untuk digunakan dalam operasi string, yaitu:

Fungsi	Keterangan
<code>strlen()</code>	Untuk mengetahui panjang string

<code>str_word_count()</code>	Untuk mengetahui jumlah kata di dalam string
<code>strpos()</code>	Untuk mengetahui posisi suatu string dalam string
<code>strrev()</code>	Untuk membalik urutan string
<code>strstr()</code>	Untuk mencari substring suatu string
<code>substr()</code>	Untuk mengambil substring dari posisi awal dan akhir dalam sebuah string
<code>trim()</code>	Menghilangkan karakter spasi di awal dan akhir string
<code>ltrim()</code>	Menghilangkan karakter spasi di awal string
<code>rtrim()</code>	Menghilangkan karakter spasi di akhir string
<code>strtoupper()</code>	Mengubah huruf menjadi huruf capital
<code>strtolower()</code>	Mengubah huruf menjadi huruf kecil (<i>lowercase</i>)
<code>str_replace()</code>	Menggantikan nilai dari beberapa bagian dari string dengan string yang lain
<code>ucwords()</code>	Mengubah huruf awal dari sebuah kata dengan huruf besar
<code>explode()</code>	Memecah sebuah kalimat berdasarkan sebuah karakter dan mengubah menjadi array

Escape Character

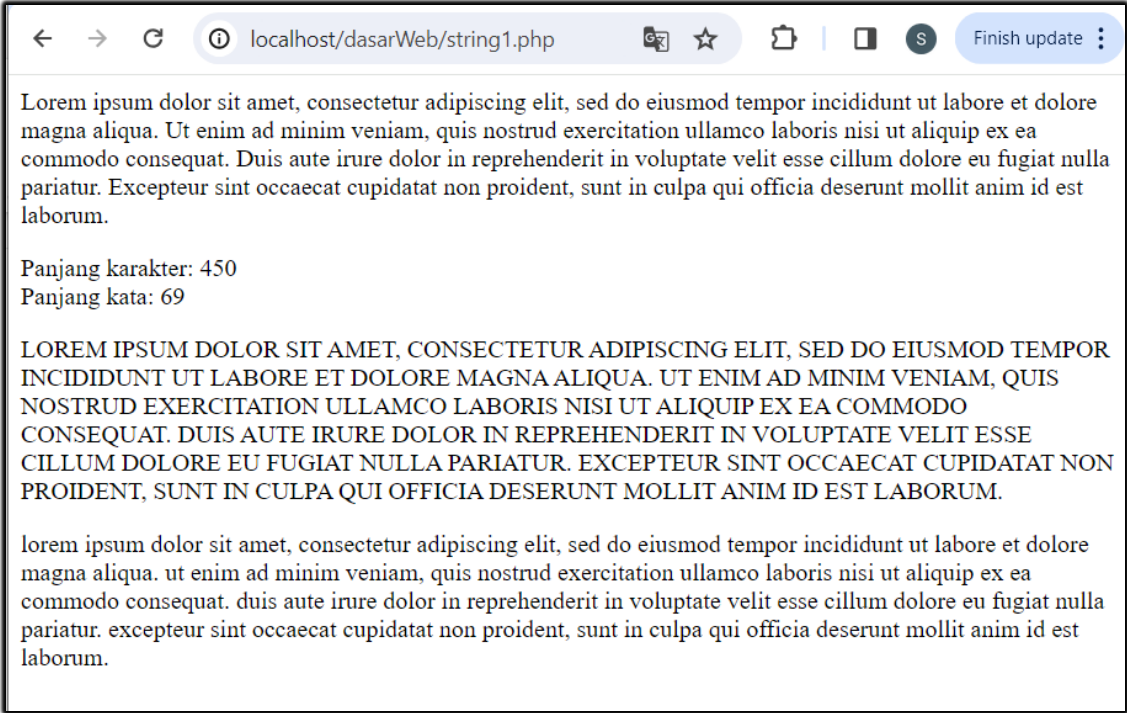

Karakter-karakter khusus yang tidak bisa ditampilkan secara langsung, melainkan harus diikuti dengan tanda \. String yang dirangkai dengan tanda petik dua akan mengganti *escape character* dengan karakter yang merepresentasikannya. Hal ini berbeda dengan string yang dirangkai dengan tanda petik satu. Yang mana ia hanya akan menampilkan apa adanya tanpa mereplace apa pun dengan apa pun (kecuali sedikit kasus).

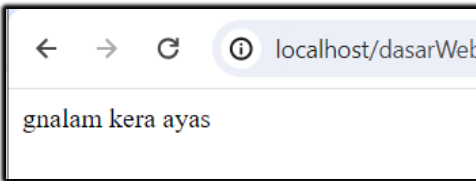
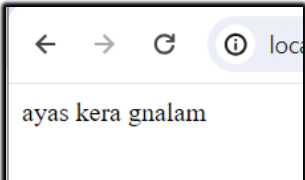
Ada pun *escape character* pada PHP adalah:

Fungsi	Keterangan
<code>\n</code>	Baris baru
<code>\r</code>	Karakter carriage-return
<code>\t</code>	Karakter tab
<code>\\$</code>	Karakter \$ itu sendiri
<code>\"</code>	Untuk menampilkan tanda petik dua
<code>\\</code>	Untuk menampilkan tanda slash \ itu sendiri

Ikuti langkah-langkah praktikum berikut ini:

Langkah	Keterangan
1	<p>Buat file <code>string1.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre><?php \$loremIpsum = "Lorem ipsum dolor sit amet consectetur adipisicing elit. Voluptatem reprehenderit nobis veritatis commodi fugiat molestias impedit unde ipsum voluptatum, corrupti minus sit excepturi nostrum quisquam? Quos impedit eum nulla optio."; echo "<p>{\$loremIpsum}</p>"; echo "Panjang karakter: " . strlen(\$loremIpsum) . "
"; echo "Panjang kata: " . str_word_count(\$loremIpsum) . "
"; echo "<p>" . strtoupper(\$loremIpsum) . "</p>"; echo "<p>" . strtolower(\$loremIpsum) . "</p>"; ?></pre>

2	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu (soal no 13)</p>  <p>Program tersebut menggunakan fungsi-fungsi PHP bawaan untuk manipulasi teks. Fungsi-fungsi tersebut termasuk <code>strlen()</code> untuk menghitung panjang karakter, <code>str_word_count()</code> untuk menghitung jumlah kata, <code>strtoupper()</code> untuk mengubah teks menjadi huruf besar, dan <code>strtolower()</code> untuk mengubah teks menjadi huruf kecil.</p>
<h3>Escape Character</h3>	
3	<p>Buat file <code>string2.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre data-bbox="284 1176 933 1473"><?php echo "Baris\nbaru
"; //soal 10.a echo 'Baris\nbaru
'; //soal 10.b echo "Halo\rDunia
"; //soal 10.c echo 'Halo\rDunia
'; ///soal 10.d echo "<pre>Halo\tDunia!</pre>"; //soal 10.e echo '<pre>Halo\tDunia!</pre>'; //soal 10.f echo "Katakanlah \"Tidak pada narkoba!\"
"; //soal 10.g echo 'Katakanlah \"Tidak pada narkoba!\"
'; //soal 10.h ?></pre>
4	<p>Dari kode program di atas, kalian bisa mengetahui perbedaan antara tanda petik dua dan tanda petik satu dari segi cara kerjanya menangani <i>escape string</i>. Amati hasil yang ditampilkan dan jelaskan hasil dari masing-masing outputnya dan apa yang dapat kalian simpulkan dari soal percobaan tersebut (soal no 14)</p> 

	Program tersebut menggunakan echo untuk mencetak teks dengan karakter khusus seperti baris baru (\n), kembali ke awal baris (\r), dan tab (\t), kutip ganda (") dan kutipan tunggal (') untuk menginterpretasikan karakter khusus atau mencetak secara harfiah.
Membalik String menggunakan perintah <i>strrev()</i> .	
5	<p>Buat file <code>string3.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre><?php \$pesan = "Saya arek malang"; echo strrev(\$pesan) . "
"; ?></pre>
6	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu (soal no 15)</p>  <p>fungsi <code>`strrev()`</code> dalam PHP untuk membalikkan urutan karakter dari string yang disimpan dalam variabel <code>`\$pesan`</code>. Hasilnya adalah string "saya arek malang" menjadi "gnalam kera ayas".</p>
8	<p>untuk membalik string per kata, ketikkan kode program berikut:</p> <pre><?php \$pesan = "saya arek malang"; # ubah variabel \$pesan menjadi array dengan perintah explode \$pesanPerKata = explode(" ", \$pesan); # ubah setiap kata dalam array menjadi kebalikannya \$pesanPerKata = array_map(fn(\$pesan) => strrev(\$pesan), \$pesanPerKata); # gabungkan kembali array menjadi string \$pesan = implode(" ", \$pesanPerKata); echo \$pesan . "
"; ?></pre>
8	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu (soal no 16)</p>  <p>Hasilnya adalah string "saya arek malang" yang dibalik menjadi "ayas kera gnalam", karena setiap kata dalam string tersebut telah dibalik urutannya.</p>

Menggabungkan HTML dan PHP

Ada dua cara menggabungkan HTML dan PHP yaitu PHP yang berada didalam HTML, dan HTML yang ada di dalam PHP.

Langkah	Keterangan
1	Cara pertama adalah php di dalam HTML. Seperti kode berikut.

	<pre> <html> <head> <title>Cara 01</title> </head> <body> <p>Tanggal Hari ini : <?php echo date("d M Y")?></p> </body> </html> </pre>
2	Pada kode diatas merupakan kode html yang berisi kode php untuk menampilkan tanggal server yang ditandai dengan tag <?php dan ?> .
3	<p>Cara kedua adalah HTML di dalam PHP. Di dalam PHP Tag HTML diperlakukan sebagai <i>string</i> yang di apit dengan tanda petik dan bisa dilakukan berbagai fungsi untuk memanipulasi <i>string</i> seperti menyambung dll.</p> <p>Contoh kode seperti pada potongan kode berikut.</p> <pre> <?php echo '<html>'; echo '<head><title>Cara02</title></head>'; echo '<body>'; echo '<p>Tanggal Hari ini : '.date('d M Y').'</p>'; echo '</body>'; echo '</html>'; ?> </pre>
4	Kode diatas mengeluarkan output sama dengan potongan kode sebelumnya. Namun yang berbeda adalah penulisan kode dimana HTML berada di dalam PHP sebagai string dan untuk menampilkannya menggunakan <i>tag echo</i> .
5	<p>Dari dua cara tersebut mana yang lebih mudah menurut kalian, kemukakan jawaban disertai dengan alasan (soal no 17)</p> <div data-bbox="284 1039 903 1279"> </div> <div data-bbox="284 1301 903 1500"> </div> <p>Menurut saya, cara pertama lebih mudah karena tidak perlu menambahkan elemen HTML tambahan, membuat kode menjadi lebih sederhana dan tidak terlalu panjang.</p>

Entities HTML

Ketika membuat konten website, tidak akan terlepas dari menggunakan karakter khusus seperti simbol copyright (©), ampersand (&), lebih kecil (<), lebih besar (>), titik koma (;) dan sebagainya. Untuk menampilkan simbol-simbol tersebut di HTML perlu menggunakan entitas karakter (*character entities*). Entitas Karakter mempunyai 3 bagian yaitu: sebuah *ampersand* (&), nama dan nomor entitas (#), dan titik koma/semicolon (;).

Sebagai contoh, untuk menampilkan simbol copyright (©) → © atau ©

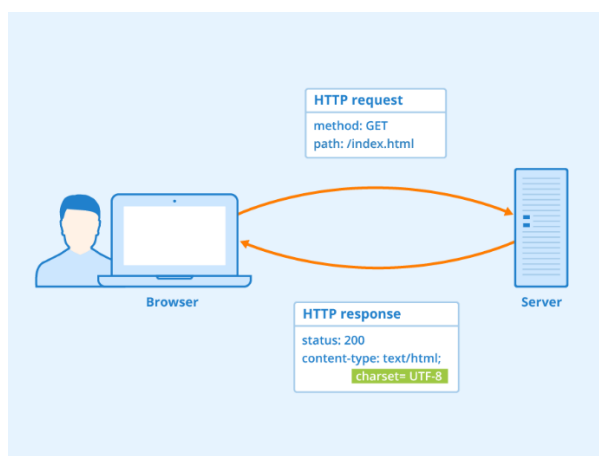
Berikut ini adalah tabel beberapa entitas yang umum digunakan:

	' digunakan untuk tanda petik tunggal '. digunakan untuk menyisipkan spasi non-breaking. © representasi dari hak cipta ©.
--	--

HTTP Header

HTTP *header* adalah sebuah data yang dikirim antara web browser dengan web server sebagai sarana komunikasi antar keduanya. Di dalam HTTP header terdapat informasi tentang bagaimana cara menangani file yang dikirim/diminta.

Siklus *request respond* untuk halaman web: Ketika kita mengakses suatu halaman web, web browser secara otomatis mengirim sebuah HTTP *request* kepada web server. HTTP *request* berisi banyak informasi, salah satunya adalah HTTP *header*. Di dalam HTTP *header* (yang dikirim pada saat proses *request*) terdapat informasi tentang file apa yang diminta (apakah file HTML, file PHP, file PDF, atau yang lain), serta berbagai info tambahan seperti jenis web browser yang dipakai, sistem operasi, dan alamat IP. Setelah sampai di web server, informasi yang ada di HTTP *header* akan dibaca dan web server mempersiapkan file-file yang diminta. Setelah itu, web server mengirim kembali file-file tersebut kepada web browser. Proses pengembalian ini dikenal juga dengan sebutan HTTP *response*.



HTTP Header

HTTP *response* ini terdiri dari 2 bagian: HTTP header dan file web. HTTP *header* berisi informasi mengenai file web yang dikirim, seperti tipe data, tanggal dikirim, nama web server, dan sistem operasi yang digunakan oleh web server. Sedangkan file web sendiri terdiri dari file HTML yang menyusun halaman web, termasuk juga file gambar (jika ada).

Sebagai analogi, jika presiden ingin berkunjung ke rumah kalian, tentu akan ada tim paspampres yang datang lebih awal. Mereka menginformasikan bahwa presiden akan datang pada jam sekian, dengan membawa sekian orang, dan informasi-informasi lain. Tim paspampres ini bisa disamakan dengan HTTP header yang datang lebih dahulu sebelum file asli dikirim. Dalam prakteknya, kita tidak menyadari adanya HTTP header, bahkan mungkin tidak pernah mendengar istilah ini. Dan itu tidak salah, karena isi dari HTTP header ditujukan kepada pemrosesan di web browser, bukan untuk pengunjung web.

Bagaimana cara melihat http header pada web browser kalian jelaskan disertai Langkah-langkahnya (Soal no 19)

1. Buka Google Chrome.
2. Buka halaman web yang ingin Anda periksa header HTTP-nya.
3. Klik kanan di area manapun pada halaman web tersebut.
4. Pilih "Inspect" atau "Inspect Element" untuk membuka Developer Tools atau Inspect Element.
5. Pergi ke tab "Network".
6. Refresh halaman web dengan menekan tombol F5 atau mengklik ikon segar di browser.
7. Lihat daftar request HTTP yang muncul di tab "Network" dan klik salah satu dari request tersebut.
8. Di bagian "Headers" (atau "Headers" di bawah "Request Headers" atau "Response Headers"), Anda akan melihat informasi header HTTP terkait dengan request tersebut.

Date and Time

Fungsi `date()` di dalam PHP digunakan untuk menampilkan tanggal dan waktu. Sintaks fungsi `date()` adalah sebagai berikut:

```
<?php
    date(format, timestamp)
?>
```

Parameter format bersifat wajib (*required*). Parameter format digunakan untuk menentukan bagaimana format tanggal dan/atau waktu yang akan digunakan. Berikut beberapa contoh karakter yang umum digunakan untuk format tanggal:

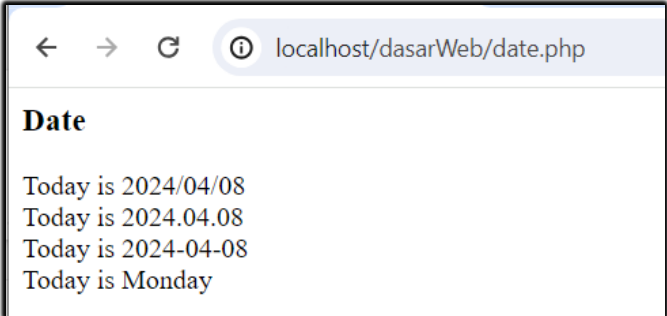
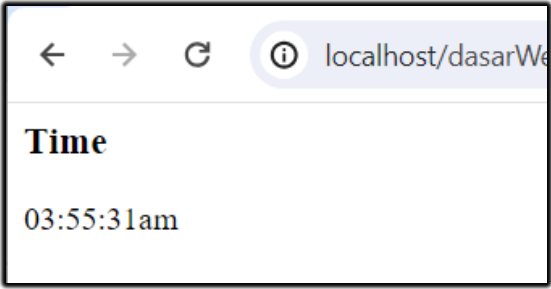
1. d, merepresentasikan hari (01 sampai 31)
2. m, merepresentasikan bulan (01 sampai 12)
3. Y, merepresentasikan tahun (dalam 4 digit)
4. l, merepresentasikan hari dalam 1 minggu

Selain menampilkan tanggal, fungsi tersebut juga menampilkan waktu. Berikut adalah beberapa karakter yang umum digunakan untuk format waktu:

1. H, merepresentasikan jam dalam format 24 jam
2. h, merepresentasikan jam dalam format 12 jam
3. i, merepresentasikan menit (00 sampai 59)
4. s, merepresentasikan detik (00 sampai 59)
5. a, merepresentasikan *ante meridiem* (am) atau *post meridiem* (pm).

Ikuti langkah-langkah berikut untuk memahami bagaimana penggunaan fungsi `date()`

Langkah	Keterangan
1	Buat file baru dengan nama <code>date.php</code> di dalam direktori <code>dasarWeb</code> , kemudian ketikkan kode berikut:

	<pre> 1 <!DOCTYPE HTML> 2 <html> 3 <head> 4 </head> 5 <body> 6 <h3> Date </h3> 7 <?php 8 echo "Today is " . date("Y/m/d") . "
"; 9 echo "Today is " . date("Y.m.d") . "
"; 10 echo "Today is " . date("Y-m-d") . "
"; 11 echo "Today is " . date("l"); 12 ?> 13 </body> 14 </html> </pre>
2	Simpan file dan jalankan kode program
3	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 19)</p>  <p>kita bisa menggunakan berbagai format date, seperti kode diatas, dan kita bisa memanggil hari dengan menggunakan karakter l, untuk tahun Y, bulan m, dan tanggal d</p>
4	<p>Buat file baru dengan nama time.php di dalam direktori dasarWeb, kemudian ketikkan kode berikut:</p> <pre> 1 <!DOCTYPE HTML> 2 <html> 3 <head> 4 </head> 5 <body> 6 <h3> Time </h3> 7 <?php 8 date_default_timezone_set("asia/jakarta"); 9 echo date("h:i:sa"); 10 ?> 11 </body> 12 </html> </pre>
5	Simpan file dan jalankan kode program
6	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 20)</p> 

	fungsi <code>date_default_time_zone_set("daerah")</code> untuk mengatur zona waktu saat ini. Karakter <code>h</code> digunakan untuk format jam dalam format 12 jam, <code>H</code> untuk format 24 jam, <code>i</code> untuk menit, <code>s</code> untuk detik, dan <code>a</code> untuk AM atau PM.
--	---

Variabel Superglobal

Variabel super global di PHP adalah variabel bawaan yang bersifat global. Variabel bawaan yang dimaksud adalah: variabel yang sudah otomatis ada tanpa perlu kita definisikan sendiri. Dan ia bersifat global dalam artian bisa kita akses dari mana pun dan kapan pun.

Variabel super global menyimpan banyak sekali data penting dan juga bermanfaat yang bisa kita gunakan dalam menyelesaikan proyek yang sedang kita kerjakan. Terdapat **9 variabel** super global di PHP.

1. Variabel \$_SERVER

Variabel yang pertama dan utama adalah variabel \$_SERVER. Ia adalah sebuah array asosiatif yang menyediakan berbagai macam informasi tentang request yang ditangkap oleh server. Data yang dimuat berupa *headers*, *paths*, lokasi skrip, dan sebagainya.

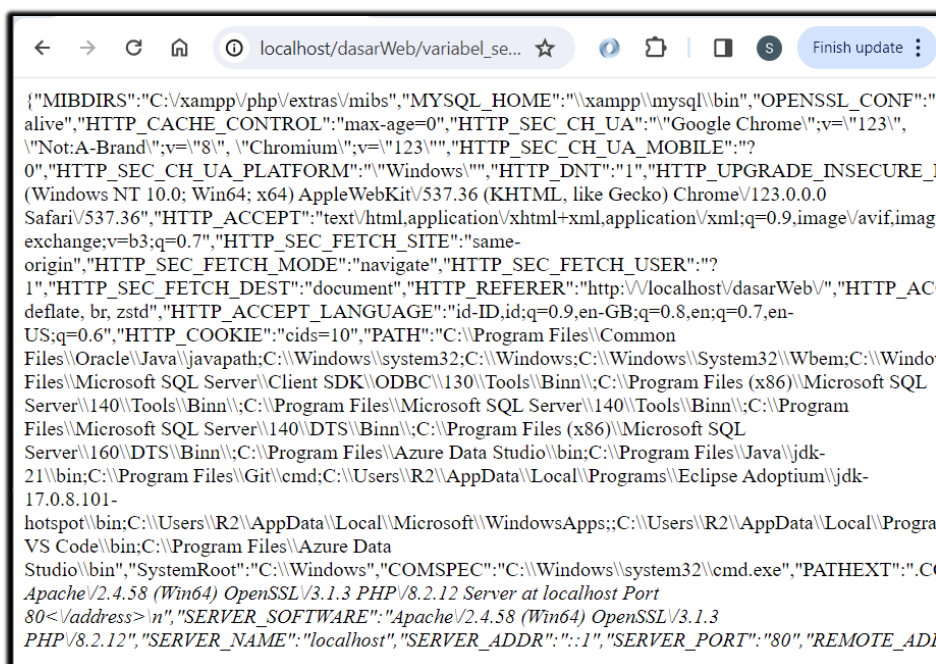
Nilai yang tersimpan pada variabel \$_SERVER disediakan oleh web server, oleh karena itu tidak ada jaminan khusus bahwa setiap web server yang kita gunakan akan memberikan semua data-data standar yang ada.

Untuk bisa mengetahui nilai apa saja yang tersedia pada variabel \$_SERVER, kita bisa mengeksekusi perintah berikut:

```
<?php  
  
echo json_encode($_SERVER);
```

Bagaimana output yang didapatkan (**Soal no 21**) sebelum dijalankan silakan install ekstensi JSON View pada link berikut ini

<https://chrome.google.com/webstore/detail/jsonvue/chklaanhfefbnpoihckbnefhakgolnmc?hl=id>



hasil diatas merupakan

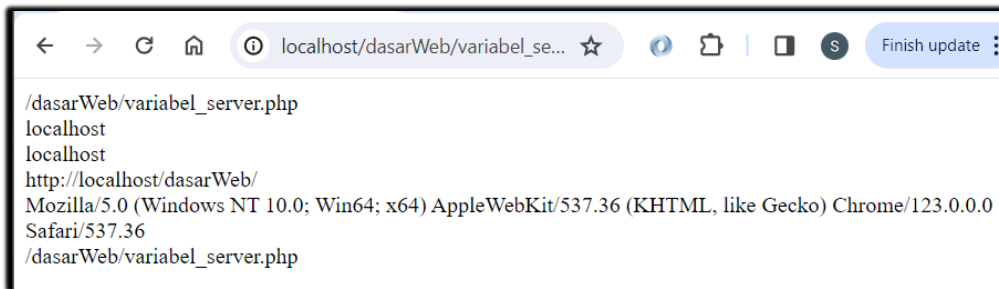
script yang berisikan nilai apa saja yang tersedia pada variable \$_SERVER


```

1 <?php
2 echo $_SERVER['PHP_SELF'];
3 echo "<br>";
4 echo $_SERVER['SERVER_NAME'];
5 echo "<br>";
6 echo $_SERVER['HTTP_HOST'];
7 echo "<br>";
8 echo $_SERVER['HTTP_REFERER'];
9 echo "<br>";
10 echo $_SERVER['HTTP_USER_AGENT'];
11 echo "<br>";
12 echo $_SERVER['SCRIPT_NAME'];
13 ?>

```

Jalankan kode program diatas kemudian jelaskan output dari masing-masing perintah echonya (Soal no.22)



Berdasarkan hasil diatas, `$_SERVER['PHP_SELF']` menyimpan nama file yang dieksekusi, `$_SERVER['SERVER_NAME']` menyimpan hostname, `$_SERVER['HTTP_HOST']` mencetak host HTTP, `$_SERVER['HTTP_REFERER']` menunjukkan halaman URL referer (atau kosong jika tidak ada), `$_SERVER['HTTP_USER_AGENT']` memberikan informasi tentang agen pengguna (user agent) browser, dan `$_SERVER['SCRIPT_NAME']` mencetak nama skrip PHP yang sedang berjalan.

Beberapa contoh data pada variabel `$_SERVER` yang akan sering dibutuhkan:

No	Variabel dan Deskripsi
1	<code>\$_SERVER['PHP_SELF']</code> Berisi informasi nama file yang sedang dieksekusi, nama file diambil berdasarkan dari dokumen <i>root</i>
2	<code>\$_SERVER['SERVER_ADDR']</code> Alamat IP server dari file yang sedang dieksekusi.
3	<code>\$_SERVER['SERVER_NAME']</code> Hostname dari server dari file php yang sedang dieksekusi. Hostname biasanya adalah nama PC yang terlihat di dalam jaringan. Jika skrip PHP dijalankan di atas Virtual Host, maka nama virtual host tersebut akan dijadikan sebagai <i>server name</i>
4	<code>\$_SERVER['SERVER_PROTOCOL']</code> Ini adalah protokol komunikasi yang sedang berjalan. Apakah HTTP atau HTTPS. Misal: 'HTTP/0.1'
5	<code>\$_SERVER['REQUEST_METHOD']</code> Berisi jenis metode <i>request</i> dari file PHP yang sedang dieksekusi. Misalkan: <code>GET</code> , <code>POST</code> , <code>PUT</code> , <code>DELETE</code> , <code>OPTIONS</code>
6	<code>\$_SERVER['QUERY_STRING']</code> Akan mengembalikan query string dari file atau skrip PHP yang sedang dieksekusi. Misalkan user mengakses halaman <code>http://localhost/halo-dunia?nama=Budi&umur=20&asal=Surabaya</code> , maka variabel tersebut akan mengembalikan nilai <code>nama=Budi&umur=20&asal=Surabaya</code> .
7	<code>\$_SERVER['DOCUMENT_ROOT']</code> Nama direktori root dari dokumen/file PHP yang sedang dieksekusi. Hasil yang dikembalikan berdasarkan dari pengaturan server.
8	<code>\$_SERVER['HTTP_HOST']</code> Mengembalikan konten dari host, seperti misalnya: header (jika memang ada).

No	Variabel dan Deskripsi
9	<code>\$_SERVER['HTTP_REFERER']</code> Halaman url yang menjadi referensi ke halaman yang sedang dieksekusi. Jika tidak ada, nilainya kosong.
10	<code>\$_SERVER['HTTP_USER_AGENT']</code> Variabel ini berisi informasi dari user yang sedang melakukan <i>request</i> konten: mulai dari browser yang digunakan, bahasa, hingga OS. Contoh nilai dari variabel ini adalah: Mozilla/4.5 [en] (X11; U; Linux 2.2.9 i586
11	<code>\$_SERVER['REMOTE_ADDR']</code> Berisi alamat IP dari user yang sedang mengakses halaman web PHP
12	<code>\$_SERVER['SCRIPT_FILENAME']</code> Nama path absolut dari file yang sedang dieksekusi.
13	<code>\$_SERVER['REQUEST_URI']</code> Alamat uri dari file yang sedang dieksekusi. Misalkan: "/php/halo-dunia".

2. Variabel \$_GET

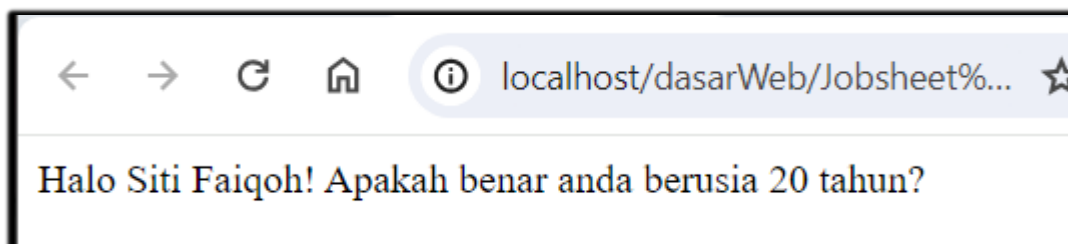
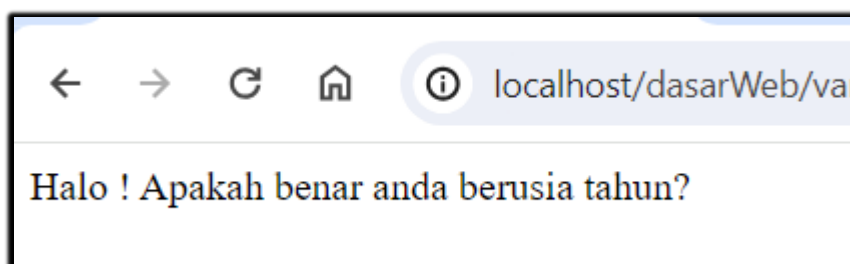
Variabel \$_GET adalah array asosiatif yang berisi nilai dari *query string*. Misalkan kita memiliki file halo-dunia.php sebagaimana berikut:

```
<?php
$nama = @$_GET['nama']; //tanda @ agar tidak ada peringatan error
                        ketika key-nya kosong
$usia = @$_GET['usia']; //tanda @ agar tidak ada peringatan error
                        ketika key-nya kosong

echo "Halo {$nama}! Apakah benar anda berusia {$usia} tahun?";
?>
```

Jika kita mengakses file tersebut dengan *query string* dibawah ini, output apa yang dihasilkan, amati dan bagaimana hasilnya (soal no 23)

<http://localhost/halo-dunia.php?nama=Elok&usia=30>



Berdasarkan hasil diatas, kita dapat menuliskan value dari variable tersebut dengan menuliskannya di Alamat url [http://localhost/dasarWeb/Jobsheet%206/variabel_get.php?nama=Siti Faiqoh&usia=20](http://localhost/dasarWeb/Jobsheet%206/variabel_get.php?nama=Siti%20Faiqoh&usia=20) seperti contoh diatas

3. Variabel \$_POST

Variabel \$_POST mirip dengan variabel \$_GET. Hanya saja data yang di-*passing* tidaklah melalui query string pada URL, akan tetapi pada *body request*. Dan *request method* yang dilakukan haruslah dengan metode **POST**.

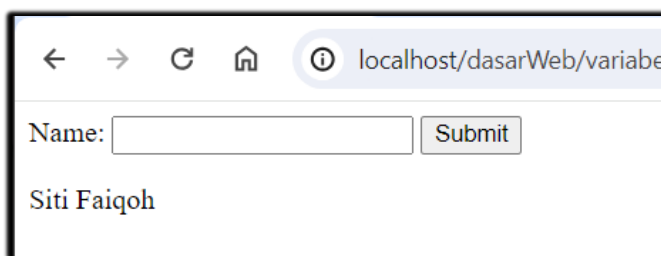
```
<html>
<body>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  // collect value of input field
  $name = $_POST['fname'];
  if (empty($name)) {
    echo "Name is empty";
  } else {
    echo $name;
  }
}
?>

</body>
</html>
```

Jalankan kode program berikut ini. Apa yang bisa kalian simpulkan dari output yang dihasilkan (Soal no 24)



Ketika kita menuliskan nama kita, maka akan menampilkan inputan tersebut, dan apabila kita tidak mengisikan apapun maka akan keluar name is empty. Kita bisa menggunakan variable \$_POST untuk mendapatkan inputan melalui body request

4. Variabel \$_SESSION

Variabel \$_SESSION adalah array asosiatif yang menyimpan data sesi pengguna. Variabel ini bisa kita gunakan untuk menyimpan user yang login pada satu sesi tertentu. Atau juga bisa digunakan untuk menyimpan data *cart* pada toko online. Secara *default*, umur sesi pada PHP adalah **1440 detik** atau **24 menit**.

5. Variabel \$_COOKIE

Mirip dengan \$_SESSION, variabel \$_COOKIE bisa kita gunakan untuk menyimpan suatu data yang berkaitan dengan user: misal informasi login, informasi *cart* pada toko online, dan sebagainya.

Bedanya, *cookie* adalah file berukuran kecil yang disimpan pada browser pengguna. File tersebut akan senantiasa dikirim setiap kali browser mengirimkan *request* ke server. Umur *cookie* umumnya lebih panjang dari pada umur sesi.

6. Variabel \$_REQUEST

Variabel `$_REQUEST` adalah array asosiatif yang menyimpan gabungan nilai dari variabel `$_GET`, `$_POST`, dan `$_COOKIE` yang kesemuanya berhubungan dengan data yang dikirim bersamaan dengan *request* user.

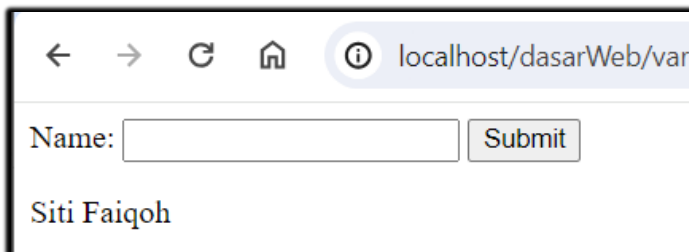
```
<html>
<body>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  // collect value of input field
  $name = $_REQUEST['fname'];
  if (empty($name)) {
    echo "Name is empty";
  } else {
    echo $name;
  }
}
?>

</body>
</html>
```

Jalankan kode program berikut ini. Apa yang bisa kalian simpulkan dari output yang dihasilkan. Dan apa bedanya dengan variable global `$POST` (Soal no 25)



Hasil sama dengan variable `$POST`, namun di variable `$REQUEST` ini akan menyimpan gabungan nilai dari variable `$GET`, `$POST`, dan `$COOKIE`

7. Variabel `$_FILES`

Variabel `$_FILES` adalah array asosiatif yang menyimpan data file yang diunggah pengguna dalam satu *request* dengan metode **POST** atau **PUT**.

8. Variabel `$_ENV`

Variabel `$_ENV` adalah array asosiatif yang berisi data tentang *environment* yang skrip PHP berjalan di atasnya. Variabel `$_ENV` disediakan oleh *shell* yang menjalankan skrip PHP, sehingga nilainya bisa bervariasi tergantung dengan sistem operasi yang digunakan.

Di dalam *framework* PHP modern seperti laravel, variabel `$_ENV` juga digunakan untuk menyimpan hal-hal yang berkaitan dengan *environment* seperti nama database, password database, dan nilai lainnya untuk melakukan konfigurasi *framework*.

9. Variabel `$GLOBALS`

Variabel `$GLOBALS` adalah array asosiatif yang menyimpan semua variabel global yang didefinisikan saat program dijalankan. Variabel `$GLOBALS` merupakan variabel super global PHP yang digunakan untuk mengakses variabel global dari mana saja dalam skrip PHP (juga dari dalam fungsi atau metode).

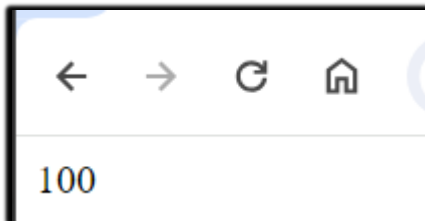
Cara menggunakan variabel super global \$GLOBALS:

```
<?php
$x = 75;
$y = 25;

function addition() {
    $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
}

addition();
echo $z;
?>
```

Bagaimana output dari kode soal di atas kemudian jelaskan! (Soal no 26)



Variable \$GOLBALS akan menyimpan semua variabel global yang didefinisikan saat program dijalankan.

Referensi:

- 1) Nixon, Robin. (2018). Learning PHP, MySQL, JavaScript, CSS & HTML: A Step-by-step Guide to Creating Dynamic Websites, 5th Edition. O'Reilly Media, Inc.
- 2) Forbes, Alan. (2012). The Joy of PHP: A Beginners's Guide to Programming Interactive Web Applications with PHP and MySQL, 5th Edition. Plum Island Publishing